

CASO #3 – PRELIMINAR #3 -Punto #2

--Consulta base

```
SELECT actor.descripcion,localesXproductor.localId
,Paises.nombre,recipientes.recipienteId,(traducciones.textoTradu) as categoria
FROM actores actor
INNER JOIN localesXproductor ON actor.actorId = localesXproductor.actorId
INNER JOIN direcciones ON localesXproductor.direccionId = direcciones.direccionId
INNER JOIN paises ON direcciones.paisId = paises.paisId
INNER JOIN recipienteXlocal ON localesXproductor.localId = recipienteXlocal.localId
INNER JOIN recipientes ON recipienteXlocal.recipienteId = recipientes.recipienteId
INNER JOIN categoriaProducto ON recipientes.categoriaId =
categoriaProducto.categoriaId
INNER JOIN traducciones ON categoriaProducto.descripcion = traducciones.traduccionId
where (recipientes.recipienteId%2)=0
ORDER BY recipientes.recipienteId ASC;
```

--Consulta de vista indexada

```
IF OBJECT_ID('dbo.vista_indexada','view') IS NOT NULL
    DROP view dbo.vista_indexada;
GO

CREATE VIEW dbo.vista_indexada
WITH SCHEMABINDING
AS
SELECT COUNT_BIG(*) AS countbig,actor.descripcion,localesXproductor.localId,
Paises.nombre ,recipientes.recipienteId,(traducciones.textoTradu) as categoria
FROM dbo.actores actor
INNER JOIN dbo.localesXproductor ON actor.actorId = localesXproductor.actorId
INNER JOIN dbo.direcciones ON localesXproductor.direccionId =
direcciones.direccionId
INNER JOIN dbo.paises ON direcciones.paisId = paises.paisId
INNER JOIN dbo.recipienteXlocal ON localesXproductor.localId =
recipienteXlocal.localId
INNER JOIN dbo.recipientes ON recipienteXlocal.recipienteId =
recipientes.recipienteId
INNER JOIN dbo.categoriaProducto ON recipientes.categoriaId =
categoriaProducto.categoriaId
INNER JOIN dbo.traducciones ON categoriaProducto.descripcion =
traducciones.traduccionId where (recipientes.recipienteId%2)=0
GROUP BY actor.descripcion,localesXproductor.localId, Paises.nombre
,recipientes.recipienteId,traducciones.textoTradu
GO

DROP INDEX vista_indexada_index on dbo.vista_indexada
GO

CREATE UNIQUE CLUSTERED INDEX vista_indexada_index
ON dbo.vista_indexada(descripcion,localId,nombre,recipienteId,categoria);
go
```

-- Consulta dinámica

```
DECLARE @sqlConsulta NVARCHAR(MAX)=N'';  
SET @sqlConsulta =  
N'SELECT actor.descripcion,localesXproductor.localId, Países.nombre  
,recipientes.recipienteId,(traducciones.textoTradu) as categoria  
FROM actores actor  
INNER JOIN localesXproductor ON actor.actorId = localesXproductor.actorId  
INNER JOIN direcciones ON localesXproductor.direccionId = direcciones.direccionId  
INNER JOIN paises ON direcciones.paisId = paises.paisId  
INNER JOIN recipienteXlocal ON localesXproductor.localId = recipienteXlocal.localId  
INNER JOIN recipientes ON recipienteXlocal.recipienteId = recipientes.recipienteId  
INNER JOIN categoriaProducto ON recipientes.categoriaId =  
categoriaProducto.categoriaId  
INNER JOIN traducciones ON categoriaProducto.descripcion = traducciones.traduccionId  
where (recipientes.recipienteId%2)=0  
ORDER BY recipientes.recipienteId ASC ';  
  
exec sp_executesql @sqlConsulta
```

Resultado/Conclusión

Utilizando una cantidad máxima de 100000 registros podemos observar los siguientes tiempos en la ejecución:

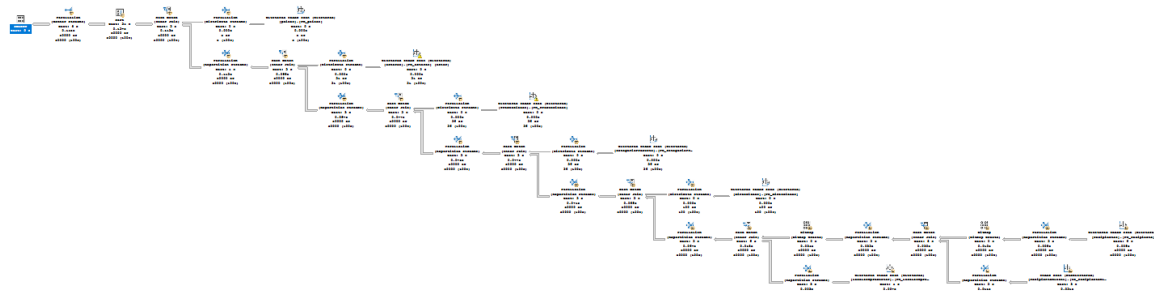
Tiempo de ejecución de la consulta base: 00:00:08

Tiempo de ejecución de la consulta indexada: 00:00:01

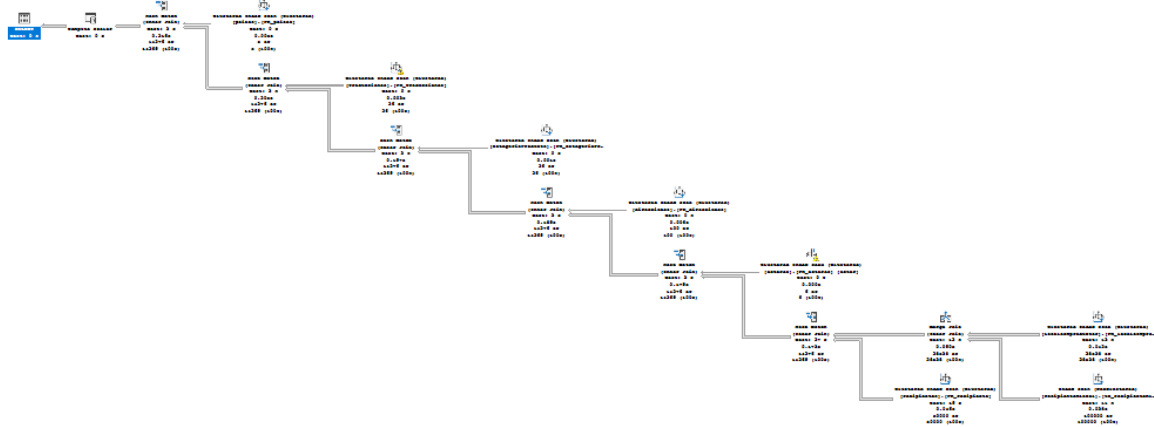
Tiempo de ejecución de la consulta dinámica: 00:00:01

Ambas tanto la consulta indexada como la dinámica optimizan el tiempo significativamente.

Execution plan vista dinámica



Execution plan vista indexada



Para esta consulta creada con el fin de retornar los recipientes y las empresas en los que se encuentran, opinamos que la mejor opción es la vista indexada, debido a que estos datos no son modificados con alta frecuencia, la vista se almacenara en una tabla separada en el disco y se actualizara automáticamente con cada nuevo registro que se ingrese acelerando las consultas que se realicen posteriormente.